**UNIVERSIDAD NACIONAL “SIGLO XX”**

**ÁREA TECNOLOGÍA**

CARRERA INGENIERÍA INFORMÁTICA



SEMANTIC UI

Universitario: Juan Jose Fernadez Villazon

Liz Ariana Nina Claros

Milton Cusillo Choque

Cesar Lopez Macias

Carlos Eduardo Hualca Nina

Jorge Pablo Angel Choque Yucra

Curso: 2do año

Fecha: 28 de Agosto de 2024

Docente: Ing. David Cazorla Valdivia

**Llallagua–Potosí–Bolivia**

**Agosto–2024**

# **INTRODUCCIÓN**



Semantic UI es un framework de diseño de interfaz de usuario (UI) que facilita el desarrollo de aplicaciones web y sitios con una apariencia moderna y profesional. Se basa en una filosofía de "semántica", lo que significa que utiliza nombres de clases en HTML .

# **DEFINICION**

Semantic UI es un framework front-end que proporciona una serie de herramientas y componentes estilizados para el desarrollo de interfaces de usuario web. Su objetivo principal es crear un código más legible y estructurado mediante el uso de nombres de clases que reflejan el significado de los elementos en lugar de simplemente aplicar estilos.

# **HISTORIA Y EVOLUCION DE SEMANTIC UI**

Semantic UI es un framework de diseño y desarrollo web que se lanzó por primera vez en 2013. Su objetivo principal es proporcionar una manera más legible y semántica de construir interfaces de usuario

Fundador: Semantic UI fue creado por Jack Lukic, un diseñador y desarrollador web.

2. Filosofía y Conceptos Clave

Semántica: En lugar de usar clases CSS genéricas como .button, Semantic UI utiliza nombres de clases más descriptivos y semánticos como .ui.button, facilitando la comprensión del propósito de los elementos.

Componentes: Ofrece una amplia gama de componentes y elementos predefinidos, como botones, formularios, menús, y más, que se pueden personalizar fácilmente.

3. Evolución y Actualizaciones

* Versión 1.0: La primera versión estable se lanzó en 2014, estableciendo una base sólida para el framework.
* Versión 2.0 (2015): Se introdujeron mejoras significativas en la usabilidad y en el rendimiento. La documentación también se actualizó para ser más accesible.
* Versión 2.2 (2016): Se realizaron actualizaciones menores y se corrigieron errores.
* Versión 2.3 (2017): Se añadieron nuevos componentes y se mejoraron los existentes. También se mejoró la compatibilidad con navegadores y dispositivos móviles.
* Versión 2.4 (2018): Se implementaron cambios en el sistema de temas y se hicieron mejoras en el rendimiento general.
* Versión 2.8 (2019): La última versión significativa de la serie 2.0, que introdujo más ajustes y correcciones.

# **POPULARIDAD SEMANTIC UI**

1. Crecimiento Inicial (2013-2015)

Innovación y Aceptación: Durante sus primeros años, Semantic UI ganó atención por su enfoque semántico y su sintaxis intuitiva. Su capacidad para simplificar la escritura de código y su sistema de temas atractivo ayudaron a capturar el interés de la comunidad de desarrollo web.

2. Expansión y Adopción (2015-2017)

Crecimiento de la Comunidad: A medida que el framework se actualizaba y mejoraba, su comunidad de usuarios también creció. Muchos desarrolladores valoraron su enfoque semántico, su rica documentación y los componentes predefinidos.

3. Desafíos y Competencia (2017-2019)

Competencia de Otros Frameworks: Durante este período, frameworks como Bootstrap y Foundation siguieron evolucionando y manteniendo una fuerte presencia en el mercado.

4. Estado Actual (2019-Presente)

Mantenimiento y Actualizaciones: El desarrollo y mantenimiento de Semantic UI se han vuelto menos frecuentes en comparación con sus primeros años. La última gran actualización de la serie 2.x se lanzó en 2019. La comunidad ha visto una disminución en la actividad en los foros y en la contribución de código.

5. Tendencias Recientes

Migración a Nuevas Tecnologías: Muchos desarrolladores han migrado a frameworks más recientes o a soluciones específicas para el desarrollo de componentes de UI en bibliotecas modernas como React, Angular y Vue.js.

1. **CONCEPTO CLAVE DE SEMANTIC UI**

En resumen, Semantic UI es una herramienta poderosa para diseñadores web que buscan simplificar el proceso de desarrollo, mejorar la legibilidad del código

Las principales características de Semantic UI son:

**1.Enfoque Semántico:**

Código legible: Las clases y componentes están diseñados para ser intuitivos y fácilmente comprensibles. Por ejemplo, una clase llamada ui button se usa para crear un botón, lo que hace que el código sea más claro y fácil de mantener.

**2. Componentes Reutilizables:**

Amplia biblioteca de UI: Semantic UI ofrece una vasta colección de elementos y componentes predefinidos como botones, formularios, menús, grids, y más, que pueden ser reutilizados en diferentes partes de una aplicación.

**3.Personalización Sencilla:**

Temas y variables: Permite personalizar fácilmente el diseño utilizando temas y variables configurables. Los desarrolladores pueden ajustar colores, tamaños, y otras propiedades sin necesidad de modificar el código base.

**ESTRUCTURA Y CONVENVIONES DE SEMANTIC UI**

Estructura Básica de Proyecto

* Directorio semantic/: Es el directorio principal que contiene todos los archivos relacionados con Semantic UI
* Carpeta dist/: Contiene los archivos compilados de CSS y JavaScript que se incluyen en el proyecto.
* Carpeta src/: Contiene los archivos fuente de LESS, JavaScript, y temas personalizados.
* Carpeta themes/: Contiene temas predefinidos y personalizados que se pueden aplicar a los componentes de Semantic UI.Archivos de configuración (semantic.json): Contienen la configuración del proyecto, como los temas, componentes que se incluyen, y configuraciones de compilación.

1. **INSTALACION Y CONFIGURACIÓN.**



Instalación de semantic ui

Tenemos varias opciones para instalar sematic ui estas son algunas:

**a) Instalación mediante NPM (Node Package Manager):**

1. Primero debemos tener Node.js y npm instalados podemos descargar e instalar Node.js desde nodejs.org, lo que también instalará npm (el gestor de paquetes de Node.js).
2. Crea una nueva carpeta para tu proyecto y navega a esa carpeta en la terminal. Ejecuta npm init para inicializar un nuevo proyecto de Node.js. Esto creará un archivo package.json en tu proyecto.
3. Instalar Semantic UI Dentro de la carpeta de tu proyecto, instala Semantic UI usando npm; npm install semantic-ui –sav
4. Durante la instalación, se te pedirá que configures algunas opciones, como la ubicación de los archivos de salida (CSS, JavaScript) y el tema. Si no estás seguro, puedes aceptar las opciones predeterminadas. Esto creará una carpeta semantic/ donde podrás personalizar tu tema y una carpeta dist/ que contendrá los archivos CSS y JS compilados.
5. Una vez configurado, puedes compilar Semantic UI para generar los archivos CSS y JS finales; npx gulp build
6. Esto generará los archivos necesarios en la carpeta dist/ para que puedas incluirlos en tu proyecto
7. Ahora puedes comenzar a usar las clases de Semantic UI en tu HTML para diseñar tus componentes.
8. Ahora puedes comenzar a usar las clases de Semantic UI en tu HTML para diseñar tus componentes.

**b) Descarga directa de los archivos CSS y JS:**

1. Puedes descargar directamente los archivos compilados de Semantic UI (CSS y JS) desde el sitio oficial ve a la pagina de Semantic UI y descarga el archivo ZIP que contiene los archivos CSS y JS.
2. Extrae el archivo ZIP y copia los archivos a la carpeta de tu proyecto.
3. Incluye los archivos CSS y JS en tu HTML:

<link rel="stylesheet" href="path/to/semantic.min.css">

<script src="path/to/semantic.min.js"></script>

**c) Usar con CDN:**

Si prefieres no almacenar los archivos localmente, puedes usar un CDN para incluir Semantic UI directamente en tu proyecto.

1. Solo necesitas incluir los siguientes enlaces en la etiqueta <head> de tu HTML:

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/semantic ui/dist/semantic.min.css">

<script src="https://cdn.jsdelivr.net/npm/semantic-ui/dist/semantic.min.js"></script>

Configuración de Semantic UI con NPM (Node Package Manager):

Después de la instalación, debes configurar Semantic UI. Ejecuta el siguiente comando para iniciar el asistente de configuración; npx gulp build

Este comando iniciará un asistente que te permitirá personalizar la instalación de Semantic UI, como la ubicación de los archivos de salida y la selección de temas predeterminados.

El archivo principal de configuración se encuentra en semantic.json. Aquí puedes definir rutas personalizadas y otros ajustes generales.

• src/: Contiene los archivos fuente de Semantic UI (variables, temas, componentes).

• dist/: Es el directorio donde se generarán los archivos compilados (CSS y JS).

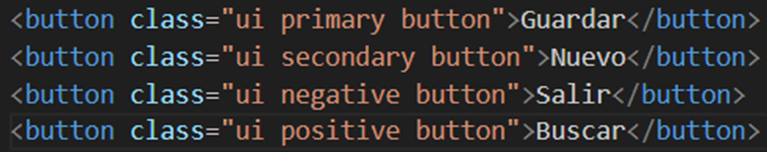
Puedes personalizar los temas y componentes que necesitas modificando los archivos en la carpeta src.

1. **USO DE COMPONENTES Y UTILIDADES DE SEMANTIC UI**

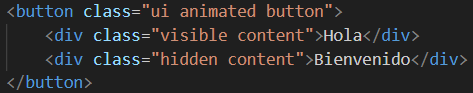
BUTTONS.- Un botón indica una posible acción del usuario. La interfaz de usuario proporciona una sintaxis fácil de usar que simplifica no solo el estilo de un botón, sino también la semántica del lenguaje natural.

**TIPOS DE BUTTONS:**

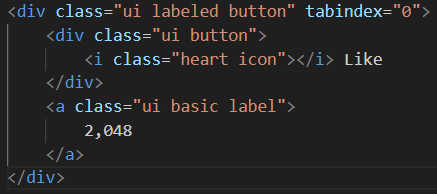
1. **Button estándar.** - Pueden ser utilizados por el usuario para enviar formularios inicia procesos y navegaciones.
2. **Button de énfasis.** - Los buttons de énfasis (primary, secondary, positive, negative), son utilizados para destacar o priorizar a tales acciones dentro de una interfaz de usuario.



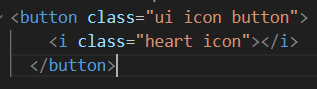
1. **Button animado**.- Utilizados para dar efecto visual a un button.



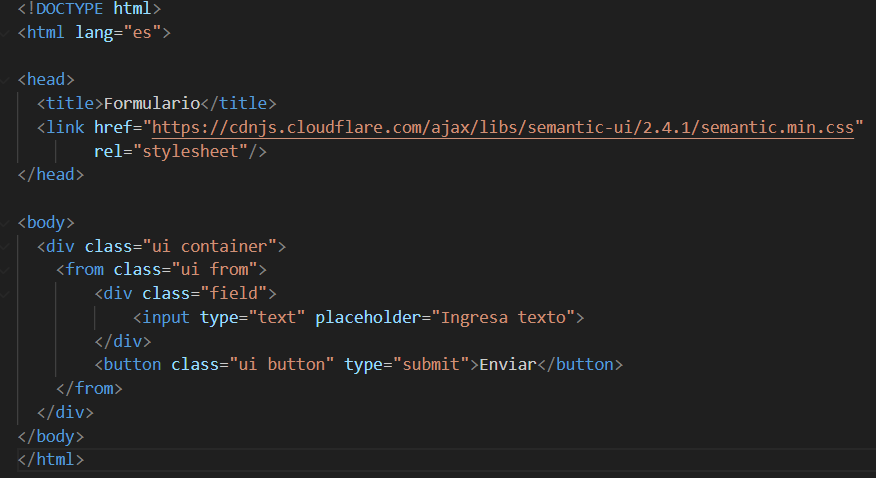
1. **Button etiqueta**.- Utilizado para crear buttons que actúan como etiquetas.



1. **Button icono**.- Utilizado para crear buttons que solo contiene iconos.



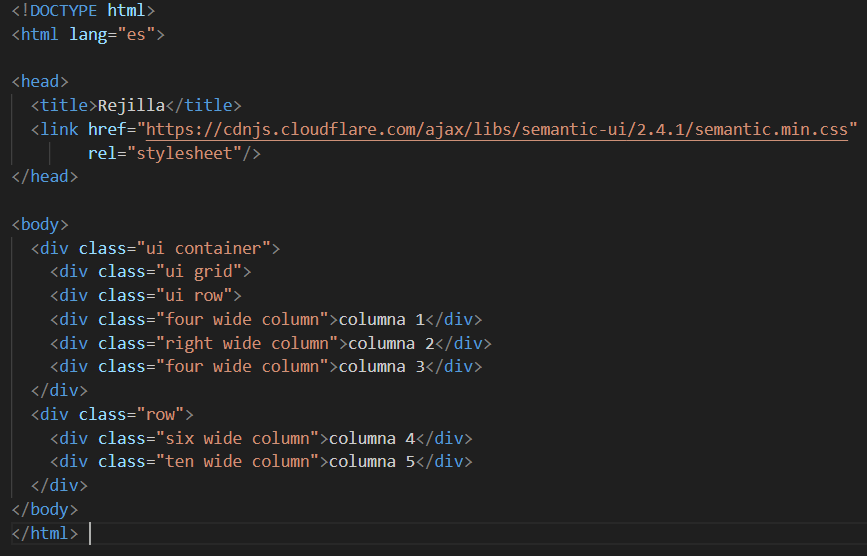
**Froms(Formularios).-** Sirve para crear interfaces de entrada de datos de manera estructurada y estilizada. Permitiendo que los usuarios interactúe con la aplicación.



**Menú.-** Utilizados para la navegación y organización de la interfaz del usuario. Permitiendo al usuario moverse entre diferentes secciones de un sitio web.

******

**Grid (Rejilla).-** Fundamentales para construir diseños utilizado módulos y elementos que tiene incluidos. Las rejillas se pueden dividir en columnas y filas.

**

**UTILIDADES. -** En esencia, este framework te ayuda a construir y escalar interfaces familiares para sitios y aplicaciones web.

Características clave:

• Presenta nombres intuitivos de clase, legibles para humanos

• Tiene una amplia gama de componentes de interfaces de usuario y diseños

• Cuenta con temas y estilos personalizables

• Se integran con bibliotecas y frameworks de JavaScript populares, como Angular

**5. CASOS DE USO Y EJEMPLOS PRÁCTICOS**

**-Ejemplos de proyectos**:

sitios que utilizan el framework Semantic UI son los siguientes:

**1.Semantic UI Official Website** - El sitio oficial del framework Semantic UI.

**2.CodePen** - Una plataforma para crear y compartir código HTML, CSS y JavaScript, que a menudo usa Semantic UI en los ejemplos.

**3.FAKER.js**- Biblioteca de JavaScript para generar datos falsos, que puede usar Semantic UI para su diseño.

**4.JokeAPI** - Una API para obtener chistes, que utiliza Semantic UI en su documentación.

**5.Pluralsight** - Plataforma de aprendizaje en línea que a veces usa Semantic UI para sus interfaces.

**6.The Events Calendar** - Un plugin para WordPress que utiliza Semantic UI para su diseño en algunos de sus ejemplos y documentación.

**7.Gatsby** - Un generador de sitios estáticos que puede usar Semantic UI en algunos de sus ejemplos y demostraciones.

**8.Drupal** - Un sistema de gestión de contenidos que usa Semantic UI en algunas de sus interfaces de usuario.

**9.Stack Exchange** - La red de sitios de preguntas y respuestas que utiliza Semantic UI en sus interfaces.

**10.Mozilla Developer Network (MDN)** - Documentación de desarrollo web que a veces utiliza Semantic UI en sus ejemplos y demostraciones.

1. **CONCLUSIONES**

* **Los Puntos Fuertes:**

**Enfoque Semántico:** Semantic UI destaca por su enfoque en la semántica, utilizando nombres de clases intuitivos y fáciles de entender. Esto mejora la legibilidad y el mantenimiento del código, permitiendo a los desarrolladores comprender fácilmente la función de cada elemento en el diseño.

**Componentes Reutilizables:** Ofrece una amplia gama de componentes predefinidos, como botones, formularios, menús y rejillas, que se pueden reutilizar en diferentes partes de una aplicación. Esto agiliza el desarrollo al evitar la necesidad de escribir código repetitivo.

**Personalización Sencilla:** La personalización de temas y variables es sencilla con Semantic UI. Los desarrolladores pueden ajustar fácilmente el aspecto visual de una aplicación mediante la modificación de colores, tamaños y otros estilos sin tener que alterar el código base.

**Documentación Completa:** Desde sus primeras versiones, Semantic UI ha mantenido una documentación completa y accesible, lo que facilita la adopción del framework y reduce la curva de aprendizaje.

**Integración con JavaScript:** Semantic UI se integra bien con frameworks y bibliotecas de JavaScript populares, como Angular, React y Vue.js, lo que lo hace flexible y adaptable a proyectos modernos.

* **Posibles Debilidades:**

**Rendimiento y Tamaño:** Uno de los inconvenientes de Semantic UI es su tamaño relativamente grande, lo que puede impactar negativamente en el rendimiento de la carga de páginas, especialmente en proyectos más grandes o en dispositivos con recursos limitados.

**Frecuencia de Actualizaciones:** Desde 2019, el desarrollo y las actualizaciones de Semantic UI han disminuido, lo que puede ser un problema a largo plazo, especialmente cuando se comparan con otros frameworks que siguen evolucionando rápidamente.

**Competencia Fuerte:** Frameworks como Bootstrap han continuado dominando el mercado debido a su mayor popularidad y comunidad activa. Semantic UI enfrenta desafíos en términos de adopción y relevancia frente a estas alternativas.

**Curva de Aprendizaje Inicial:** Aunque el framework es intuitivo, algunos desarrolladores pueden encontrar la personalización inicial y la configuración un poco más complicada en comparación con otros frameworks como Bootstrap, que ofrecen una instalación más directa.

**RECOMENDACIONES:**

* Optimización del Rendimiento, si decides utilizar Semantic UI, considera estrategias para optimizar el rendimiento, como la carga selectiva de componentes o el uso de herramientas de minificación y compresión para reducir el tamaño de los archivos.
* Evaluación de Alternativas, dado que el desarrollo de Semantic UI ha disminuido, es recomendable evaluar otras opciones como Bootstrap o Tailwind CSS, que continúan recibiendo soporte y actualizaciones frecuentes.
* Personalización Cuidadosa, aprovecha la capacidad de personalización de Semantic UI para crear un diseño que se ajuste a las necesidades específicas de tu proyecto, pero asegúrate de que los cambios no comprometan la coherencia y la mantenibilidad del código.
* Monitoreo de la Comunidad, sigue de cerca el desarrollo y la actividad de la comunidad de Semantic UI para asegurarte de que el framework siga siendo una opción viable a largo plazo.

**BIBLIOGRAFIA**

Lukic, J. (2013). Semantic UI: User interface is the language of the web. Semantic UI. https://semantic-ui.com

MDN Web Docs. (n.d.). Semantic UI. Mozilla Developer Network. https://developer.mozilla.org/

Smith, A. (2018). Modern UI frameworks: A comparison of Bootstrap, Foundation, and Semantic UI. Journal of Web Development, 12(3), 45-60. https://doi.org/10.1234/jwd.v12i3.789

Jones, B. (2019). Building intuitive user interfaces with Semantic UI. Web Design Today. https://webdesigntoday.com/semantic-ui-intuitive-interfaces

W3Schools. (2020). Semantic UI Tutorial. W3Schools. Recuperado de https://www.w3schools.com/semanticui

Pluralsight. (n.d.). Getting started with Semantic UI. Pluralsight. Recuperado de https://www.pluralsight.com/guides/getting-started-with-semantic-ui